Searching for Globally Optimal Functional Forms for Interatomic Potentials Using Genetic Programming with Parallel Tempering

A. Slepoy, M. D. Peters, A. P. Thompson*
Sandia National Laboratories
Albuquerque, New Mexico 87185

Molecular dynamics and other molecular simulation methods rely on a potential energy function, based only on the relative coordinates of the atomic nuclei. Such a function, called a force field, approximately represents the electronic structure interactions of a condensed matter system. Developing such approximate functions and fitting their parameters remains an arduous, time-consuming process, relying on expert physical intuition. To address this problem, a functional programming methodology was developed that may enable automated discovery of entirely new force field functional forms, while simultaneously fitting parameter values. The method uses a combination of genetic programming, Metropolis Monte Carlo importance sampling and parallel tempering to efficiently search a large space of candidate functional forms and parameters.

The methodology was tested using a non-trivial problem with a well-defined globally optimal solution: a small set of atomic configurations was generated and the energy of each configuration was calculated using the Lennard-Jones pair potential. Starting with a population of random functions, our fully-automated, massively parallel implementation of the method reproducibly discovered the original Lennard-Jones pair potential by searching for several hours on 100 processors, sampling only a minuscule portion of the total search space. This result indicates that, with further improvement, the method may be suitable for unsupervised development of more accurate force fields with completely new functional forms.

PACS numbers: 66.20.+d, 05.60.-k,
Keywords: force field, interatomic potential, genetic programming

I. INTRODUCTION

Classical molecular dynamics [MD] and other molecular simulation methods have found broad application in many areas of science and technology, including the nanoscale design of synthetic materials, and understanding structure and function of biomolecules. All such simulations require accurate and computationally efficient force fields, functions that calculate energy and forces of the system from a set of atomic coordinates.

Currently used force fields represent extensive work in invention and validation. The functional forms of force fields are chosen based on physical intuition; their parameters (i.e. multiplicative constants, exponents, etc.) are then adjusted to achieve a good fit to selected properties. Each such process, done manually, can take many man-years of highly qualified labor, and often meets with failure.

With increasing interest in predicting the detailed quantitative behavior of real condensed matter systems, there is a growing need for more complicated functional forms. Such functional forms must be better able to represent the intricacies of the, possibly changing, electronic structure. It is difficult, if not impossible, to develop an intuition for the relationships between the complicated functions and the ever-expanding training set of material data.

An automated functional form optimization method is clearly called for. We wish to extend the search beyond parameter-only optimization and use unsupervised machine algorithms to refine the functional form as well. To do this, we have adopted and combined state-of-the-art ideas from the field of evolutionary computing. In the following section we introduce the concept of a force field, including the Lennard-Jones pair potential that is the subject of the current study. In Section III we describe our search algorithm in detail. In Section IV we describe how our method successfully discovered the Lennard-Jones pair potential. We conclude with a discussion of how the method may be applied to the development of more complex force fields.

II. BACKGROUND

Molecular simulation methods use a force field to describe the potential energy surface of a group of atoms. The force field can most generally be written as an expansion in multi-body interaction terms. Each n-body term is a functional expression that returns the energy due to a group of n atoms, with the total potential energy summed over all available groups and all multi-body terms,

$$E_{conf}(\mathbf{r}_1, ..., \mathbf{r}_N) = \sum_{<i,j>} u_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_{<i,j,k>} u_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + ... \quad (1)$$

The sums are over all spatially proximate groups of n atoms i.e. the interactions between atoms separated by more than some finite distance is assumed to be negligible. The resulting scalar is the potential energy of the

system; spatial derivatives of the potential energy with respect to individual atomic coordinates give the force vectors acting on the respective atoms. The individual multi-body terms are functions of the coordinates of the n atoms, with parameter values that depend on the identities of the atoms. Hence a full definition of a particular force field requires specification of a set of multi-body functional forms as well as the parameters values that are to be used for each combination of atom types.

Most commonly used force fields carry only a few terms of the full expansion. For example, the Coulombic potential can be represented by a particular 2-body term, dependent only on the scalar distance between pairs of atoms:

$$u_{2,C}(\mathbf{r}_i, \mathbf{r}_j) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \tag{2}$$

$$r_{ij} = \|\mathbf{r}_j - \mathbf{r}_i\|. \tag{3}$$

The parameter q_i here denotes the electrostatic charge of particle i. It should be noted that because of the slowly-decaying form of the Coulombic potential, the interaction between particles at large separations is usually calculated using special techniques such as Ewald summation.

A system of particles, interacting via harmonic springs, can also be represented using just a 2-body term:

$$u_{2,H}(\mathbf{r}_i, \mathbf{r}_j) = k_{ij} r_{ij}^2, \tag{4}$$

where the parameter k_{ij} is the spring constant between particles i and j. Though both of these force field functional forms use the same order term and describe interactions between pairs of atoms in terms of the relative distance, they are quite distinct in functional form and use different parameters. Another purely 2-body force field functional form that is widely used for condensed matter systems is the Lennard-Jones pair potential[1]:

$$u_{2,LJ}(\mathbf{r}_i, \mathbf{r}_j) = 4\epsilon \left\{ \left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right\}, \tag{5}$$

where ϵ and σ are energy and length parameters, respectively. This simple function represents both the Pauli exclusion repulsion between valence shell electrons at short separations as well as the attractive interaction between induced dipoles at larger separations. In the current work we use the Lennard-Jones force field function to test our methodology.

To meet increasingly stringent requirements for the quantitative description of condensed matter systems, force field functional forms have necessarily become more complex and include higher-order terms in the expansion. For example, cubic crystalline solids represented by a purely 2-body force field functional form have shear elastic constants that are exactly equal, $C_{12}/C_{44} = 1$, whereas in most metals this ratio is substantially greater than unity. Overcoming this limitation was a key factor in the widespread adoption of the embedded-atom method force field functional form for metals[2]. In general, identification of appropriate force field functional forms represents a great challenge to the molecular simulation community. Force field development remains an art with very few successful practitioners[3,4].

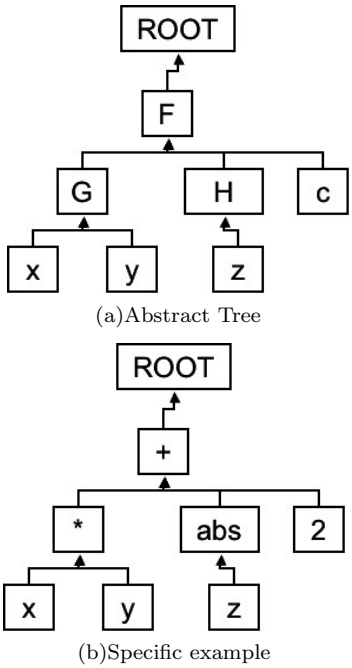


FIG. 1: (a) Graphical tree representation of a genetic program $F(G(x, y), H(z), c)$. The arrows indicate the data flow sense. (b) A particular instance of the program $xy + \|z\| + 2$.

While a great need exists for a more fully automated approach, most algorithmic research has been confined to parameter optimization of existing functional forms[5–10]. Several groups have avoided the issue of functional forms entirely, relying on purely numerical representations of the potential energy surface using tabulated values or spline functions[11,12]. Our strategy is intermediate between these two extremes. We still wish to rely on a functional form, but we would like the choice of functional form to be part of the automated optimization process. In the following section we describe in detail a new systematic method for unsupervised search for optimal force field functional forms.

III. METHOD

Our method uses an operator tree representation of the force field functional form, and a novel optimization algorithm which combines an evolutionary optimization approach with parallel tempering. The definition of the fitness function, while fairly straightforward, is also described for completeness.

A. Tree Representation

The genetic programming methodology was originally developed by Koza[13,14] to enable automatic generation of computer programs. The method represents a function as a tree of elementary operators, as depicted in Figure 1(a). The nodes of the tree can be of three basic types: elementary operators, input variables, and constant parameters. In Figure 1(b), the " + " operator is an elementary operator, x is an input variable, and 2 is a constant parameter. Input variables refer to function arguments, labeled by a variable name. The depth level of a node denotes the number of steps to the root node. The node containing " $*$ " lies at depth level 2.

The trees are used to compute whatever quantities are required by the fitness function. In the current study, each tree was used to compute the energy of a pair of atoms separated by a given distance. Constant parameters were restricted to integers on the range $[-20, 20]$. Elementary operators were restricted to the following set of simple arithmetic operators: addition $+$, subtraction $-$, multiplication $*$, division $/$, exponentiation $\hat{}$, and absolute value $|\ |$. More generally, the genetic programming representation can use any elementary operators that can be expressed as a computer program with well-defined inputs and outputs.

B. Fitness

The fitness of a tree is a numerical measure of how well the output of a tree reproduces the data in a training set. In the work presented here, the training set consists of 10 configurations of 10 particles, placed in a three-dimensional domain. An energy was computed for each configuration using the Lennard-Jones pair potential given by Eq. 5,

$$E_{conf} \;=\; \sum_{<i,j>} u_{2,LJ}(\mathbf{r}_i, \mathbf{r}_j). \qquad (6)$$

The configuration domain had dimensions of $3\sigma \times 3\sigma \times 3\sigma$, where σ is the Lennard-Jones distance parameter. The particles were placed randomly in the domain, but no particles were allowed to come closer than 0.5σ. All pair distances in the range $0.7\sigma < r < 2.0\sigma$ were recorded and used to compute the target configuration energy, taking periodic images into account. The values of σ and ϵ were set to unity. This resulted in about 60 pair distances per configuration. The training set consisted of the set of pair distances and the corresponding configuration energy. By varying the random number seed, we generated four independent training sets.

The fitness of a tree was determined by comparing each configuration energy calculated by the tree with the configuration energy calculated by the Lennard-Jones pair potential. The configuration energy for a particular tree was computed as:

$$\tilde{E}_{conf} \;=\; \sum_{<i,j>} \tilde{e}_{ij}(r_{ij}), \qquad (7)$$

where, $\tilde{e}_{ij}(r)$ is the value generated by the tree given the input value r. The fitness of the tree was then defined to be the negative square error, averaged over all configurations,

$$F \;=\; -\frac{1}{N_{conf}} \sum_{conf=1}^{N_{conf}} \left(\tilde{E}_{conf} - E_{conf}\right)^2, \qquad (8)$$

where the negative sign was required to have the fitness increase with decreasing error.

The purpose of this study was to test whether the approach was capable of discovering compact force field functions that accurately represent the potential energy surface of condensed matter systems. Typically, information about the energy surface is obtained from quantum mechanics calculations that estimate the energy of small configurations of atoms. The training set used in the current study was relatively simple but it nonetheless captured some of the characteristics of real training sets used for force field development.

C. Evolutionary Optimization Algorithm

Evolutionary optimization algorithms operate on populations of objects using bio-mimetic principles of natural selection[14]. In each new generation, parent objects form offspring, and the fittest offspring form the new population. In the case of genetic programming, the high-level strategy builds a population of random operator trees and iteratively refines the population using tree evolution operators to generate new offspring trees, which may or may not be admitted into the new generation.

Evolution proceeds in three stages: generation, mutation, and testing. The first stage produces N_t new trees from N_t old trees. The second stage randomly mutates some of these trees. The third stage compares the fitness of each of the N_t new trees with that of the old trees and admits either the new one or the old one into the new generation.

In the generation stage, each of the N_t new trees is created either by pass-through or crossover, with equal probability. Pass-through selects the fittest tree the first time, the second fittest tree the second time, and so on, copying them into the new population without modification. Cross-over creates a new tree by combining two parent trees selected from the old population (Figure 2(c)). Each of the two parents is chosen via tournament selection: four trees are chosen with equal probability from the old population, and the tree with highest fitness is selected. To perform crossover using the two parents, a depth level is selected for the first parent, with the restriction that the node is not the root or at maximum

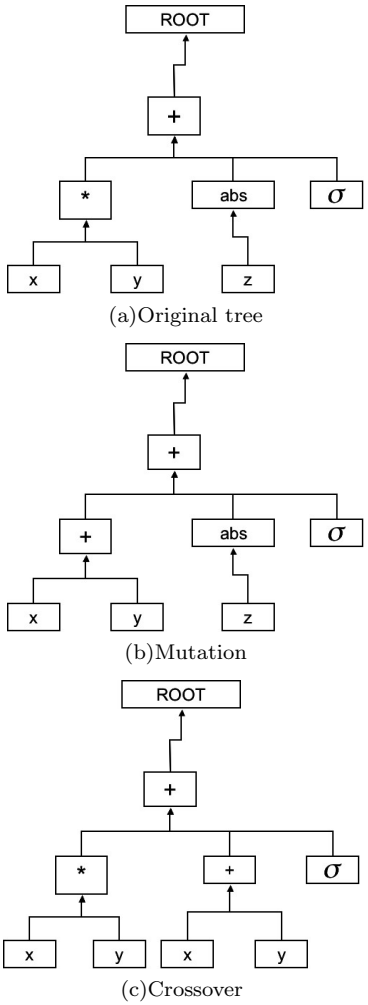


FIG. 2: Tree evolution operators are illustrated. The original tree (a) is evolved to tree (b) by a mutation of the $*$ operator to the $+$ operator. Tree (c) is generated by cross-over of trees (a) and (b); the $\|z\|$ branch of (a) is replaced by the $x + y$ branch of the tree (b).

depth. A depth level is then selected for the second parent, with the same restrictions. A randomly chosen sub-tree, rooted at the selected depth level, is then cut from the first parent, and replaced with a randomly chosen sub-tree, rooted at the selected depth level in the second parent, producing one new child tree containing parts of both parents.

In the mutation stage (Figure 2(b)), each new tree is either mutated or left unchanged, with equal probability and without regard for fitness or how the tree was created. The node is selected with equal probability from all nodes, meaning that there is a higher probability to select a node near the leaves than to select a node near the root. This is done to give a preference for small adjustments to the parameter nodes rather than drastic changes to the entire functional form. The sub-tree rooted at the selected node is deleted and a new random sub-tree is generated. Thus, the new generation is composed of four

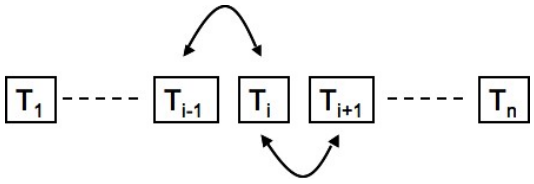


FIG. 3: Graphical representation of a parallel tempering algorithm with replicas marked with their individual temperatures.

different categories of tree, occurring in roughly equal numbers: those made by crossover alone, those made by crossover and mutation, those made by mutation alone, and those retained from the previous generation.

In the testing stage, the old trees, ordered by fitness, and the new trees, which are in the order they are created, are compared pairwise. Each old tree is compared against the new tree in the same position in the list, and one or the other is chosen for the new generation. If the fitness of the new tree is higher than that of the old one the new tree is always chosen. If the new tree fitness is lower, then it is chosen with the Boltzmann probability:

$$P_{acc} \;=\; \min\left\{1, \exp\left[\left(F^{new} - F^{old}\right)/T\right]\right\}, \qquad (9)$$

where F^{old} and F^{new} are the fitnesses of the old and new trees, respectively (8). T is the effective temperature, a non-physical parameter, used to improve search efficiency (see Section III D below). At high effective temperatures, most new trees are accepted, even if their fitness is poor, favoring efficient exploration of the search space. At low effective temperatures, only trees with improvements or small decreases to fitness are accepted, favoring incremental improvement. By separating these two activities into different populations using the parallel tempering technique described next, we achieve simultaneous exploration and incremental improvement.

D. Parallel Tempering

Parallel tempering (PT) was originally introduced by Swendsen and Wang[15] to deal with the local traps of a spin glass energy surface. Their technique uses N replicas of the system, each at a different temperature, and exchanges partial state information between replicas. The fundamental idea is to use the high-temperature replicas to sample the system phase space at a coarse level with the low-temperature replicas refining the states in local traps. In this way, a hybrid of local and global sampling can be achieved. Later changes to the method replace partial information exchange with a complete state swap. Many parameters of the method have come under scrutiny[16].

In our method, a single replica consisted of a population of operator trees. After each generation, each population attempted to exchange one tree with its left neighbor in the temperature space (Figure 3) and then one tree

with its right neighbor. The trees to be swapped were selected with equal probability from the entire population of the respective replicas. A swap was accepted with a probability based on the relative Boltzmann weights of the two trees,

$$P_{acc} = \min\{1, \exp\left[(1/T_i - 1/T_{i+1})\left(F_{i+1} - F_i\right)\right]\}, \quad (10)$$

where F_i is the fitness of the tree selected in population i at a temperature T_i. If the swap had the effect of moving the fitter tree to the lower temperature, the swap was always accepted. Otherwise, the acceptance probability decreased exponentially with increasing fitness difference.

IV. RESULTS

The calculations were run on a cluster of 100 AMD Opteron 2.2 GHz processors with Quadrics interconnects. We ran two different parallel tempering optimizations, with either $N_t = 10,000$ or $N_t = 50,000$ individual trees in each replica. In both cases, we used 200 replicas with temperatures distributed logarithmically from 0.1 to 10 (the units of temperature were the same as those of the fitness function i.e. ϵ^2). All trees were required to have minimum depths of 3 and maximum depths of 4.

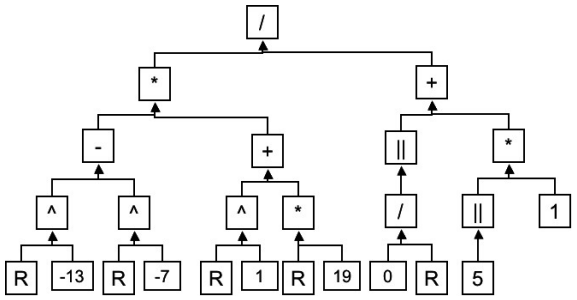
For the runs with 10,000 individuals per replica, each generation required about 100 seconds. For runs with 50,000 individuals, the time per generation was about 5 times longer. Most of this time was spent in the evaluation of configuration energies.

The results of the runs were stochastic, both due to the initial conditions and the optimization method, and so we see a distribution of behaviors. However, all but one of the runs successfully found an arithmetic equivalent of the original target function. Three such algebraic equivalents are displayed in Figure 4.

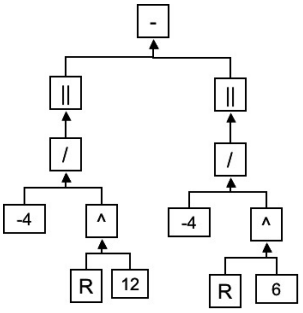
Figure 5(a) shows the average square error for the fittest tree in each generation. The dashed lines indicate four independent runs with $N_t = 10,000$ individuals per replica, each run using different initial populations and different training sets. The solid lines indicate four independent runs with $N_t = 50,000$ individuals per replica, each using different initial populations and the same training sets used for the first four runs.

Of the four independent runs with $N_t = 10,000$, three of them successfully found arithmetic equivalents of the Lennard-Jones pair potential. The residual average square error of approximately $10^{-9}\epsilon^2$ can be attributed to finite machine precision. The fourth run failed to find an arithmetic equivalent, even after 400 generations. However, it did find several functions that were good approximations to the Lennard-Jones pair potential, but have quite different functional forms (see the tree shown in Fig. 4(d)).

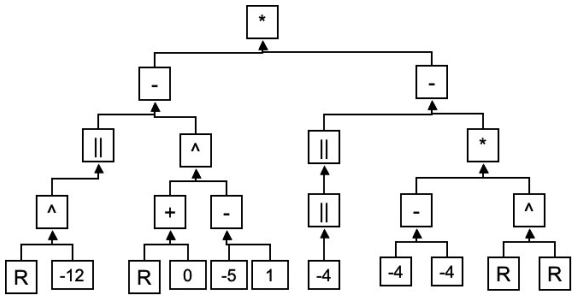
In the case of the larger populations, $N_t = 50,000$, all four runs found arithmetic equivalents and did so in



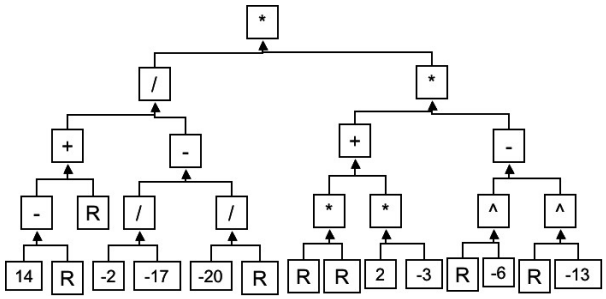
(a)Lennard-Jones equivalent tree 1



(b)Lennard-Jones equivalent tree 2

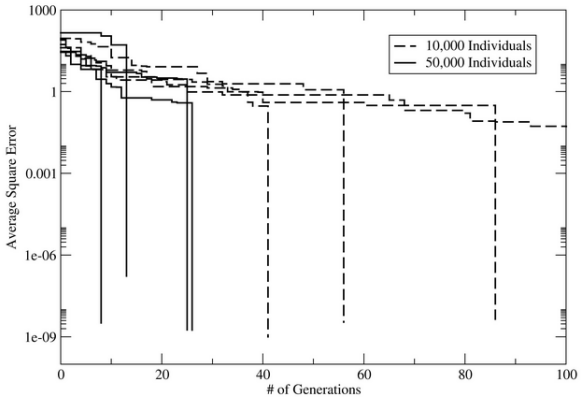


(c)Lennard-Jones equivalent tree 3

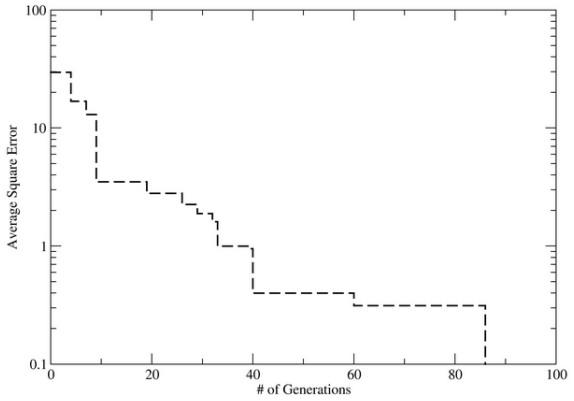


(d)Near-miss tree with high fitness value.

FIG. 4: Lennard-Jones equivalent trees and a near miss. The trees shown in (a), (b), and (c) are algebraically equivalent to the Lennard-Jones form. The tree shown in (d) produces a function that replicates the Lennard-Jones pair potential numerically over the range of interest with error less then 10%. The function represented in (a) is $\frac{[r^{-13}-r^{-7}][r^1+19r]}{\|\frac{0}{r}\|+\|5\|\times 1}$, (b) is $\|\frac{-4}{r^{12}}\| - \|\frac{-4}{r^6}\|$, and (c) is $(\|r^{-12}\| - (r+0)^{-5-1}) \times (\|\|\| - 4\|\|\| - (-4-(-4)) \times r^r)$, all of which reduce to $4(r^{-12} - r^{-6})$. The function represented in (d) reduces to $\frac{7}{10+r/17}(6 - r^2)(r^{-12} - r^{-5})$, a good approximation to the Lennard-Jones pair potential



(a)Average square error for a series of runs.



(b)Average square error for a single run.

FIG. 5: Convergence of the average square error (negative fitness) of the fittest tree in each generation, (a) all eight runs. There are four independent runs with N_t = 10,000 (dashed lines) and four independent runs with N_t = 50,000 (solid lines). In all but one case an algebraic equivalent of the Lennard-Jones pair potential was found. (b) One of the N_t = 10,000 runs in more detail.

roughly five times fewer generations. Apparently, adding more individuals speeds up the search rate proportionately.

Figure 5(b) shows how one of the $N_t = 10,000$ runs progressed. Initially fitness improved quite steadily, until a good approximation to the exact Lennard-Jones pair potential was found. After this point, further improvement occurred only sporadically. Eventually, the arithmetic equivalent of the target entered the population.

V. DISCUSSION

The problem of finding the correct tree in the space of all possible trees of a given size is made difficult by the sheer size of the search space. Our algorithm, starting with an initial total population of $\sim 10^7$ trees, in most cases found the global optimum in less than 100 generations. This provides an upper bound for the number of trees surveyed equal to 10^9.

To assess the efficiency of the search method, we can compare this figure to the total number of possible trees of depth 4. For simplicity, we ignore the unary operator $|\ |$ and ignore trees that are not maximal. Then, for M binary operators, and a maximum depth of K, the lower bound of the number of possible trees at operator-only level is given by:

$$N_{op} = M \times M^2 \times ...M^{2^{K-1}} = \prod_{k=0}^{K-1} M^{2^k}.$$

The number of leaves holding integer constants on the range $[-P, P]$ or the input variable is given by

$$N_{val} = (2P+2)^{2^K}.$$

In our case $K = 4, M = 5$, and $P = 20$, and so the total number of possible trees is,

$$N_{tree} = N_{val} \times N_{op} = 42^{16} \times 5^{15} = 2.9 \times 10^{36}.$$

Clearly, finding the global optimum in such a large space is a challenging problem, especially given that the space is likely to be very heterogeneous, with many local traps, and roughness on many scales. Exhaustive enumeration of all possible trees is not computationally feasible. The above estimates indicate that the ratio of the total number of trees to the number of surveyed trees is at least 10^{27}. We were able to accomplish the task of finding the needle in this particular haystack by sampling only a minuscule fraction of all possible trees. We succeeded in our unsupervised automated search for a functional form partly because we used a very robust optimization method and relatively large computational resources. We also speculate that the large-scale fitness landscape may have a convex funnel-like shape. This would tend to facilitate gradual progress towards the global optimum, despite the numerous local traps.

It is important to emphasize that while the manufactured test problem used in this study had a known solution, it nonetheless was representative of many potential energy surfaces where accurate functional forms are unknown. We deliberately chose a training set that closely resembled data generated by quantum density functional theory energy calculations, because this type of data is often used to develop new force fields.

Given the success of the method in the current study, we now intend to apply the method to some of the many condensed matter systems for which existing force fields

have been found deficient. A material of particular interest is bulk germanium. Despite its chemical similarity to silicon, for which several accurate force fields exist, no force field of comparable accuracy has been found for germanium. In particular, all of the commonly used germanium force fields fail to provide an adequate description of the solid-liquid coexistence line[17].

In order to use this new methodology to develop new force fields for systems of practical interest, we will have to extend the existing implementation in several ways. The allowable range for parameter values will be extended to the set of all (machine-representable) real numbers. The space of operator trees will be extended to include 3-body interactions, and eventually to even more complicated forms. The training sets will be extended to include additional properties such as elastic constants, structural properties, and forces on individual atoms. All of this extra complexity is necessary, but it also presents two risks. Firstly, the search space may become so large that it may be difficult or impossible to find a good fit to the training set. Secondly, even if a good fit is found, the resultant force field may not be sufficiently accurate for atomic configurations not included in the training set. This second problem is related to the important issue of transferability: can a force field be used to predict properties other than those to which it was fit?

Our methodology opens access to a vast search space of generic functional forms. We warn however that unrestrained use of generality can lead to an uncontrollable explosion in the size of this space. An understanding of the symmetries of a particular physical problem can be instrumental in reducing the size of the search space and therefore improving the efficiency of the method. By judicious combination of physical intuition and algorithmic flexibility, significant control can be achieved in tuning the method to produce novel functional forms for atomistic force fields.

Acknowledgments

The authors are grateful to John B. Aidun and John D. Siirola for a critical reading of the manuscript, and Richard J. Pryor for the helpful discussions on the fine points of genetic programming. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin company, for the United States Department of Energy under contract No. DE–AC04–94AL–85000.

* Electronic address: aslepoy,mpeters,athomps@sandia.gov

[1] J. E. Lennard-Jones. Cohesion. *Proceedings of the Physical Society*, 43:461 – 482, UK 1931.

[2] M. S. Daw, S. M. Foiles, and M. I. Baskes. The embedded-atom method: a review of theory and applications. *Material Science Reports*, 9(7/8):251 – 310, MAR 1993.

[3] D. W. Brenner. The art and science of an analytic potential. *Physica Status Solidi B*, 217(1):23 – 40, JAN 2000.

[4] J. W. Ponder and D. A. Case. Force fields for protein simulations. *PROTEIN SIMULATIONS*, 66:27, 2003.

[5] A. Globus, M. Menon, and D. Srivastava. Javagenes: evolving molecular force field parameters with genetic algorithm. *Computer Modeling in Engineering and Sciences*, 3(5):557 – 74, OCT 2002.

[6] T. R. Cundari and W. T. Fu. Genetic algorithm optimization of a molecular mechanics force field for technetium. *INORGANICA CHIMICA ACTA*, 300:113 – 124, APR 2000.

[7] J. Hunger, S. Beyreuther, G. Huttner, K. Allinger, U. Radelof, and L. Zsolnai. How to derive force field parameters by genetic algorithms: Modelling tripod-Mo(CO)(3) compounds as an example. *EUROPEAN JOURNAL OF INORGANIC CHEMISTRY*, (6):693 – 702, JUN 1998.

[8] J. Hunger and G. Huttner. Optimization and analysis of force field parameters by combination of genetic algorithms and neural networks. *Journal of Computational Chemistry*, 20(4):455 – 71, MAR 1999.

[9] Junmei Wang and P. A. Kollman. Automatic parameterization of force field by systematic search and genetic algorithms. *Journal of Computational Chemistry*, 22(12):1219 – 28, 2001.

[10] G. A. Kaminski, R. A. Friesner, J. Tirado-Rives, and W. L. Jorgensen. Evaluation and reparametrization of the OPLS-AA force field for proteins via comparison with accurate quantum chemical calculations on peptides. *Journal of Physical Chemistry B*, 105(28):6474 – 87, Jul 2001.

[11] F. Muller-Plathe. Coarse-graining in polymer simulation: From the atomistic to the mesoscopic scale and back. *CHEMPHYSCHEM*, 3(9):754 – 769, SEP 2002.

[12] S. Izvekov, M. Parrinello, C. J. Burnham, and G. A. Voth. Effective force fields for condensed phase systems from ab initio molecular dynamics simulation: a new method for force-matching. *Journal of Chemical Physics*, 120(23):10896 – 913, Jun 2004.

[13] J. R. Koza. Genetic programming: routine, human-competitive, high-return machine intelligence. *AISB Quarterly*, (114):5 –, 2003.

[14] John R. Koza, Matthew J. Streeter, and Martin A. Keane. Routine human-competitive machine intelligence by means of genetic programming. SPIE conference. In Bruno Bosacchi, David B. Fogel, and James C. Bezdek, editors, *Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation VI*, volume 5200 of *Proceedings of SPIE*, pages 1–15. SPIE, San Diego, California, 2003.

[15] R. H. Swendsen and J. S. Wang. Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters*, 57(21):2607 – 9, NOV 1986.

[16] D. J. Earl and M. W. Deem. Parallel tempering: theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910 – 16, 2005.

[17] S. J. Cook and P. Clancy. Comparison of semi-empirical potential functions for silicon and germanium. *Physical Review B*, 47:7686 – 7699, USA 1993.